

# Utilización de CLAPACK para resolver sistemas de ecuaciones lineales mediante paralelismo y optimización de memoria

Germán Harvey Alférez Salinas - Universidad de Montemorelos, México  
Edward Mauricio Alférez Salinas - Universidad de Nova, Lisboa

---

Resumen: Actualmente muchos problemas de la ciencia y la tecnología son modelados y solucionados usando sistemas de ecuaciones lineales. Sin embargo, este tipo de sistemas pueden llegar a ser en situaciones de uso real muy complejos y por lo tanto, altamente demandantes en capacidad de procesamiento computacional y memoria.

El presente artículo muestra una solución de bajo costo a este tipo de problemas, basada en el uso de paralelismo y optimización de memoria, usando computadores personales y librerías de dominio público como CLAPACK.

Palabras clave: CLAPACK, librería de dominio público, niveles de memoria, computación paralela, sistemas de ecuaciones lineales, atributos de desempeño.

Abstract: Many problems in science

and technology are modeled and solved using systems of linear equations. However, this kind of problems can become very complex and therefore, impose especial processing and memory demands.

This paper presents a low-cost solution based on the use of parallel computation and memory optimization using personal computers and public domain libraries such as CLAPACK.

## Introducción

Los sistemas de ecuaciones lineales se utilizan para resolver muchos problemas de la ciencia y la ingeniería (1). La solución de sistemas de ecuaciones lineales de  $n$ -dimensiones en general requiere operaciones aritméticas  $O(n^3)$  (2) que pueden consumir una gran cantidad de procesamiento computacional y un alto grado de movimiento de datos en los niveles de memoria cuando se

trata de problemas muy complejos en donde intervienen muchas ecuaciones (3). Esta situación se traduce en altos costos en la utilización de recursos computacionales, tanto en tiempo como en dinero.

Con el fin de resolver esta situación se ha planteado la utilización de computación paralela para tratar con problemas grandes y obtener una mejor relación costo-beneficio al ejecutar programas en una cantidad más corta de tiempo (4-5).

La presente investigación sirve como apoyo para investigadores tanto del sector académico como empresarial en el uso de paralelismo y la optimización en la utilización de niveles de memoria en la solución de sistemas de ecuaciones lineales de gran tamaño utilizando librerías de dominio público como CLAPACK (6).

Este artículo está estructurado de la siguiente manera: la Sección 1 describe a CLAPACK junto con otras librerías como LAPACK [7-9] que fueron base para su diseño y construcción; la Sección 2 presenta el desarrollo y los resultados de un experimento en el que se utilizaron cuatro computadores de escritorio con el fin de obtener conclusiones acerca del desempeño de CLAPACK; finalmente, la Sección 4 presenta conclusiones, y la Sección 5 delinea el trabajo futuro.

## 1. LAPACK y CLAPACK

LAPACK (Linear Algebra PACKage) es una librería de subrutinas escritas en el lenguaje

de programación Fortran77 (10) para resolver la mayoría de los problemas que ocurren en álgebra lineal. LAPACK se distingue por la utilización de paralelismo con memoria compartida y porque fue diseñada para ser eficiente en un amplio rango de computadores modernos de alto desempeño.

LAPACK provee muchas rutinas para la solución de sistemas de ecuaciones lineales. Entre los problemas específicos que LAPACK puede resolver están los problemas de mínimos cuadrados lineales, problemas de valor propio, problemas de valores singulares y factorización de matrices LU, Cholesky, QR, SVD, Schur y Schur generalizado. Asimismo soporta matrices densas y banda, pero no matrices generales esparcidas. Por otra parte, provee funcionalidades similares para matrices reales y complejas, tanto en precisión simple como doble.

El objetivo original del proyecto LAPACK consistió en hacer que las librerías EISPACK (11) y LINPACK (12) corrieran eficientemente en vectores de memoria compartida y en procesadores paralelos. En estos tipos de máquinas, LINPACK y EISPACK son ineficientes debido a que sus patrones de acceso a memoria hacen caso omiso de las jerarquías de memoria de las máquinas y por consiguiente, gastan mucho tiempo moviendo datos. LAPACK soluciona este problema mediante la reorganización de los algoritmos que utilizan operaciones basadas en bloques, tales como la

multiplicación de matrices. Estas operaciones de bloques pueden ser optimizadas para cada arquitectura con el fin de responder a la jerarquía de memoria y así proveer una forma para alcanzar una alta eficiencia en diversas máquinas modernas.

Las rutinas de LAPACK están escritas de tal manera que la computación sea ejecutada mediante BLAS (Basic Linear Algebra Subprograms) (13, 14). LAPACK fue diseñado desde el principio para explotar el Nivel 3 de BLAS – un grupo de especificaciones para subprogramas de Fortran (15) que realizan varios tipos de multiplicación de matrices y la solución de sistemas triangulares con múltiples lados de mano derecha (6).

Por otra parte, la librería CLAPACK fue construida utilizando la utilidad f2c (6) que convierte código escrito en Fortran en código de C. Para crear CLAPACK, fue necesario ejecutar la librería completa de LAPACK escrita en Fortran 77 [10] a través de f2c para obtener código C, y después se modificó para mejorar la lectura de su código fuente. Es importante notar que todas las rutinas de LAPACK están incluidas en CLAPACK. Además, el manejo de bloques y de los BLAS de LAPACK son equivalentes al de CLAPACK.

El objetivo principal de CLAPACK consiste en proveer LAPACK a usuarios que no tienen acceso a compiladores Fortran o que no están familiarizados con la gramática del lenguaje. Además, gracias a que

CLAPACK tiene versiones tanto para MS Windows como para sistemas operativos como LINUX, puede ser utilizada por una gran cantidad de usuarios. Por esta razón y por ser una librería de dominio público de alto desempeño para resolver sistemas de ecuaciones lineales mediante paralelismo y optimización de memoria, CLAPACK fue elegida para los experimentos de la presente investigación.

## 2. Desempeño de CLAPACK

Este capítulo describe los resultados de un experimento que muestra aspectos relacionados con el desempeño de CLAPACK versión 3.0 en cuatro computadores de escritorio con MS Windows XP al solucionar sistemas de ecuaciones lineales de gran tamaño. En el experimento se utilizaron las siguientes rutinas:

SGETRF (precisión simple) y DGETRF (precisión doble) que computan una factorización LU de una matriz general A n-por-n utilizando pivoteo parcial con intercambio de filas. La factorización tiene la forma  $A=P*L*U$ , donde P es la matriz de permutación, L es la matriz triangular inferior y U es la matriz triangular superior.

SPOTRF (precisión simple) y DPOTRF (precisión doble) que computan la factorización de Cholesky de una matriz A real simétrica definida positiva. La factorización tiene la forma  $A=U^H*U$ , si  $UPLO='U'$ , ó  $A=L*L^H$ , si  $UPLO='L'$ , donde U es una matriz triangular superior y L es una matriz triangular inferior. La

factorización tiene la forma:  $A = U^{**T} * U$ , si UPLO = 'U', ó  $A = L * L^{**T}$ , si UPLO = 'L', donde U es matriz triangular superior y L es una matriz triangular inferior.

Con el fin de tomar los tiempos en la experimentación se utilizaron las funcionalidades XLINTIMS y XLINTIMD de CLAPACK. La figura 1 muestra los resultados al utilizar la rutina SGETRF en matrices de orden ascendente. La velocidad se midió en MFLOPS (MFLOPS es un acrónimo que significa Mega Floating point Operations Per Second – Millón de Operaciones de Punto Flotante por Segundo). Los MFLOPS son una medida del desempeño computacional [15]) y se tomaron valores para cinco números de bloques en donde NB significa Número de Bloques. NB igual a 1 significa que no se utilizó un algoritmo de bloques, en otras palabras, no se subdividió la matriz original en sub-bloques (no hubo paralelismo).

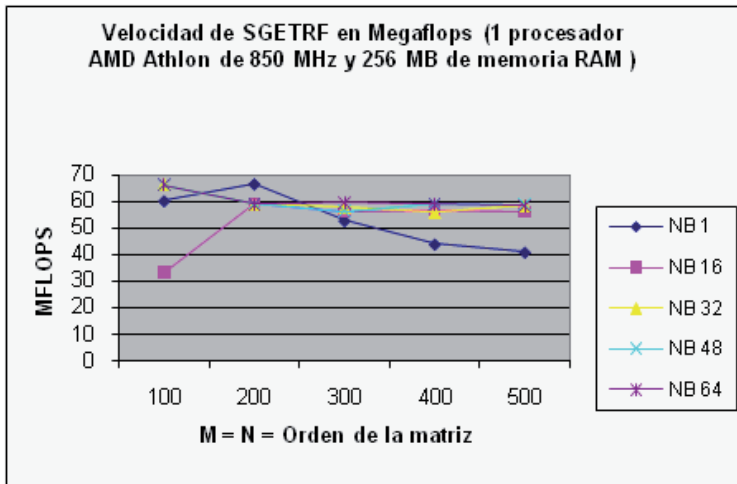


Figura 1. Desempeño de SGETRF en MFLOPS (1 procesador AMD Athlon de 850 MHz y 256 MB de memoria RAM)

En la figura 1 se observa que al utilizar la rutina SGETRF se obtienen excelentes resultados con números de bloque mayores que 1, aun al aumentar el orden de la matriz. Por el contrario, cuando el NB es igual a 1 se observa una disminución gradual al aumentar el orden. Esta tendencia también se observó en los resultados de los otros tres computadores y en precisión doble.

Por el contrario, existen rutinas en las que se ven resultados diferentes.

Por ejemplo, la figura 2 muestra que los mejores resultados de desempeño fueron obtenidos con un número de bloque igual a 1. Este hecho se repitió en los cuatro computadores tanto en precisión simple como doble. Con respecto a esto se puede decir que para obtener mejores resultados en CLAPACK, se debe conocer el número de bloque óptimo para ser utilizado en el experimento dependiendo de la arquitectura del computador.

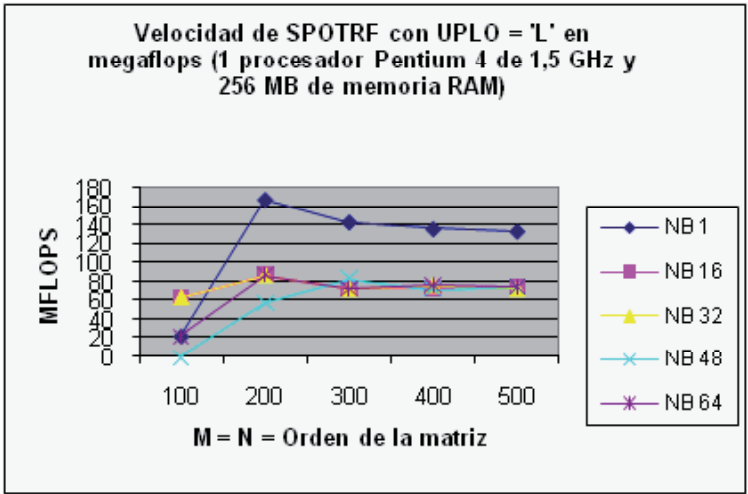


Figura 2. Desempeño de SPOTRF con UPLO = 'L' en MFLOPS (1 procesador Pentium 4 de 1,5 GHz y 256 MB de memoria RAM)

### 3. Conclusiones

En primer lugar, el presente artículo mostró que el paralelismo y la optimización del manejo de datos en los niveles de memoria de computadores personales son herramientas para la solución de sistemas de ecuaciones lineales a partir de métodos directos (en contraste con métodos iterativos que intentan resolver un problema mediante aproximaciones sucesivas a la solución) y utilizando librerías de dominio público como una opción que representa ahorros en tiempo y dinero.

Se comprobó que el desempeño de CLAPACK en computadores personales es bueno con matrices de un alto orden dependiendo del número de bloque que se utilice.

Asimismo, se encontró que la estrategia clásica de la computación

de “Divide y vencerás” tiene cabida en el marco de la división por sub-bloques.

Se observó que se puede hacer uso de las utilidades XLINTIMS y XLINTIMD que ofrece CLAPACK con el fin de conocer de antemano cuál es el número óptimo de tamaño de bloque que se debe utilizar en la arquitectura en donde se harán los experimentos, y así escoger algoritmos que utilicen únicamente el nivel 2 ó 3 de BLAS.

Además, se encontró que las velocidades de ejecución de las rutinas de CLAPACK dependen del orden de la matriz y de la arquitectura, y se demostró que no siempre se obtienen los mejores resultados con un número de bloque de grado mayor.

Por último, se demostró que el desempeño de las rutinas que

utilizan precisión simple y doble no difiere demasiado. Por lo tanto, se puede hacer utilización de la precisión doble sin temer que el tiempo de ejecución crezca en un alto grado.

#### 4. Trabajo futuro

Continuando el estudio de librerías de dominio público, vamos a realizar experimentos con la librería ScaLAPACK (Scalable LAPACK) (16-21). Estos resultados, junto con los resultados obtenidos con CLAPACK, servirán de base para el diseño de un “cluster” de computadores que se planea construir en la Facultad de Ingeniería y Tecnología, de la Universidad de Morelia, México.

#### Referencias

- Narayanan, Mysore. (n.d.). *Importance of linear algebra in engineering design methodology*. Retrieved April 21, 2009, from: <http://www.siam.org/meetings/la03/proceedings/narayanan.pdf>
- Krishna, H., & Morgera, S. (1984). *Linear complexity fast algorithms for a class of linear equations*. En: IEEE International Conference on Acoustics, Speech, and Signal Processing. Los Alamitos. IEEE Computer Society Press.
- NASA. (n.d.). *Solution of the system of linear equations*. Retrieved April 21, 2009, from: <https://www.nas.nasa.gov/Research/Reports/Techreports/1994/HTML/node34.html>
- Akl, Selim. (1997). *Parallel computation. Models and Methods*. New Jersey. Prentice Hall.
- Dongarra, Jack, et al. (1996). “ScaLAPACK: linear algebra software for distributed memory architectures”. En: *Parallel Computers. Theory and Practice*. Capítulo 7. Los Alamitos, California. IEEE Computer Society Press.
- Netlib Repository at UTK and ORNL. (n.d.). *CLAPACK (f2c'ed version of LAPACK)*. Retrieved April 19, 2009, from: <http://www.netlib.org/clapack/>
- Anderson, E., et al. (1995). *LAPACK users' guide*. Segunda ed., Philadelphia. Society for Industrial and Applied Mathematics.
- Treffitz, Christian, et al. (2002). *Una introducción al uso de LAPACK*. Universidad EAFIT. Medellín, Colombia: Universidad EAFIT.
- Netlib Repository at UTK and ORNL. (n.d.). *LAPACK – Linear Algebra PACKage*. Retrieved April 19, 2009, from: <http://www.netlib.org/lapack/>
- Rowlands, Helen. (1998). *Professional programmer's guide to Fortran77*. Retrieved April 21, 2009, from: <http://rsusu1.rnd.runnet.ru/develop/fortran/prof77/prof77.html>
- Netlib Repository at UTK and ORNL. (n.d.). *EISPACK*. Retrieved April 19, 2009, from: <http://www.netlib.org/eispack/>
- Netlib Repository at UTK and ORNL. (n.d.). *LINPACK*. Retrieved April 19, 2009, from: <http://www.netlib.org/linpack/>
- Netlib Repository at UTK and ORNL. *BLAS Frequently Asked Questions*

- (FAQ). Retrieved April 19, 2009, from: <http://www.netlib.org/blas/faq.html>
- Blackford, Susan. (1997). *BLAS*. Retrieved April 19, 2009, from: <http://www.netlib.org/scalapack/slug/node13.html#SECTION04132000000000000000>
- Webopedia. (n.d.). *FLOPS*. Retrieved April 21, 2009, from: <http://www.webopedia.com/TERM/F/FLOPS.html>
- Dongarra, Jack, et al. "ScaLAPACK: linear algebra software for distributed memory architectures". En: *Parallel Computers. Theory and Practice*. Capítulo 7. Los Alamitos, California. IEEE Computer Society Press. 1996.
- Blackford, Susan. (1997). *Software components*. Retrieved April 19, 2009, from: <http://www.netlib.org/scalapack/slug/node11.html#400>
- Blackford, Susan, et al. (1997). *ScaLAPACK user's guide*. Retrieved April 19, 2009, from: [http://www.netlib.org/scalapack/slug/scalapack\\_slug.html](http://www.netlib.org/scalapack/slug/scalapack_slug.html)
- Blackford, Susan. (1997). *ScaLAPACK quick reference guide*. Retrieved April 19, 2009, from: <http://www.netlib.org/scalapack/slug/node183.html#SECTION04111000000000000000>
- Dongarra, Jack. (n.d.). *Overview of ScaLAPACK*. Retrieved April 19, 2009, from: <http://www.netlib.org/utk/people/JackDongarra/pdf/over-scalapack.pdf>
- Dongarra, Jack, et al. (2001). *LAPACK Working note 93. Installation guide for ScaLAPACK*. Versión 1.7. Agosto, 2001. (Versión PDF disponible en <http://www.netlib.org/lapack/lawns/lawn93.ps>).